REMARKS

Claims 1-30 are currently pending. Applicant respectfully requests reconsideration of this application.

Claims 1, 3, 8, 12, 19, 21, and 27 have been amended. Claims 2, 9, 13, 20, and 28 have been cancelled without prejudice. No claims have been added.

Therefore, claims 1, 3-8, 10-12, 14-19, 21-27, 29, and 30 are now presented for examination.

Claim Amendments

Incorporation of Subject Matter of Existing Dependent Claims

To expedite prosecution, claims 1, 8, 12, 19, and 27 have been amended, with the subject matter of all of such amendments being drawn from existing dependent claims. Thus, the claims presented herein have already been examined in this prosecution.

The respective dependent claims 2, 9, 13, 20, and 28 have then been cancelled. The remaining amendments to claims 3 and 21 are solely technical amendments to address the changes in dependencies that are required as a result of the cancellation of claim 2 and 20.

No other changes are made to the claims, and thus all elements of the claims presented here have already been examined.

Claim Rejection under 35 U.S.C. §103

Klimenko, et al. in view of Burokas, et al. and Haskins

The Examiner rejected claims 1-30 under 35 U.S.C. 103 (a) as being unpatentable over U.S. Patent 5,974,547 of Klimenko, et al. (hereinafter referred to as "*Klimenko*") in view of U.S. Patent 6,954,852 of Burokas, et al. (hereinafter referred to as "*Burokas*")

and in further view U.S. Patent 6,240,169 of Haskins (hereinafter referred to as "*Haskins*")

Claim 1 – As amended herein, claim 1 is as follows:

1. A method comprising:
   requesting a memory address region and network boot load data from a server;
   receiving the network boot load data and a designated memory region from the server;
   loading the network boot load data into the designated memory region;
   running the network boot load data;
   jumping to a designated memory region for an operating system; and
   initializing the operating system.

The Examiner has modified the previous rejection of this claim to rely upon the combination of *Klimenko*, *Haskins*, and the newly cited element of *Burokas*. However, it is respectfully submitted that the combination of references cited by the Examiner do not teach or reasonably suggest the elements of claim 1. It is submitted that *Haskins* and *Burokas* do not teach or reasonably suggest the claim limitations that are missing from the primary reference *Klimenko*, and thus the references, separately or together, cannot teach or suggest the elements of the claims.

***Klimenko* Reference** – As was discussed in the previous response, *Klimenko* involves a form of network booting of an operating system to a client computer. In particular, *Klimenko* provides for storing an image of a client hard drive, including an operating system. This image is then accessed by a hard disk emulation process. As

described in the Summary of *Klimenko*, there is a continuous client hard disk emulation that exists throughout the boot process. (*Klimenko*, col. 3, lines 50-57)

As has been discussed previously, what is described in *Klimenko* is a different kind of process than is described in claim 1. While *Klimenko* refers to "network booting" of an operating system, it is referring to the location of the operating system and applications, not to the boot load data. *Klimenko* utilizes a network to present an image of client's hard drive, and then uses this image to load the operating system. *Klimenko* does not address the obtaining boot data – the data required to boot a system – from the network, but is rather involved in the question of whether the operating system and applications are stored.

It is thus again submitted that *Klimenko* addresses a different issue than Claim 1. Claim 1 relates to "requesting a memory address region and network boot load data from a server", "receiving the network boot load data and a designated memory region from the server", and then "loading the network boot load data into the designated memory region". Claim 1 then provides for "running the network boot load data", "jumping to a designated memory region for an operating system", and "initializing the operating system". In contrast, *Klimenko* does not describe a system in which a memory address region and network boot load data are requested and received from a server, and in which the network boot load data is loaded into the designated memory region. *Klimenko* instead describes a system in which a bootloader is conventionally downloaded and used for the boot process.

In the current Office Action, it is stated that "Klimenko does not explicitly teach that the network boot load data (boot code) is provided from a server. In other words, the

network boot load data (boot code) of Klimenko is not stored on a server. The network boot load data (boot code) of Klimenko is stored on a NIC of the client PC." It is respectfully submitted that this does not address the key points of the issue. The issue is not simply the storage of the boot code or the receipt from a server, but process of "requesting a memory address region and network boot load data from a server". *Klimenko* does not provide for either the request for the memory address region or the network boot load data. In claim 1, upon the request being made the network boot load data and a designated memory region are received from the server, and the network boot load data is loaded into the designated memory region. Unless these elements are present, the references are not relevant to what is occurring in claim 1.

The Examiner cites to certain portions of *Klimenko* with regard to the claim elements, with the main portion being included in the following:

> Once a user has powered-up client PC 10, as symbolized by block 420, the stored ROM BIOS in the client PC is loaded into RAM 332 (see FIG. 3) of the client PC from which that code is then executed by the PC. This operational mode is denoted by block 425 shown in FIGS. 4A and 4B. At this point, as symbolized by block 430, the client PC is not aware of its IP address. The client PC then reads the boot code from a ROM situated on the NIC (or alternatively on the motherboard of the client PC itself) into RAM 332 and then executes that code--this operational mode denoted by block 450. In response to this code, the client PC will broadcast, as symbolized by line 432, a BootP (or DHCP) request packet over the network to elicit a response from a BootP (or DHCP) server. Illustratively, server 50 contains BootP server 232. This packet contains the hardware address of the NIC. ...

(*Klimenko*, col. 9, line 56 through col. 10, line 4) (emphasis added) As indicated, the boot code is read from a ROM on the NIC of the motherboard of the client PC. This does not include any request for a memory address region or network boot load data from a server. In fact, the description indicates that the boot code is being read from read only memory. Thus, not only does *Klimenko* not teach or suggest the elements of the claim, the described system operates in such a way that the elements of the claim would be incompatible with *Klimenko*. The reference suggests reading boot data from a read only memory, which is incompatible with a request for network boot load data – the concepts do not make any sense together. The operation of *Klimenko* is further shown in the following text in describing Figure 2A:

> As shown, client PC 10 contains LAN adapter (also commonly referred to as a network interface card--NIC) 360. Each such NIC carries a unique physical hardware address, referred to as a media access control (MAC) address, through which that card can be uniquely addressed on a network. An illustrative MAC address is "00A024Baf9a5". Each NIC also contains internal read only memory 362 that stores boot code 364, which contains a BootP client process. Though this code is usually stored within the NIC, as shown here, this code could alternatively be implemented within a PC ROM BIOS (basic input output system) located on a motherboard of the client PC. With the boot code stored in the NIC, as shown, and read into memory of the PC on power-up and executed, the client PC establishes a network connection, through network 30 and connections 20 and 40, with remote server 50 for remotely booting of the client PC. Server 50 contains, to the extent relevant to the present invention, TCP (transmission control protocol) servers 230, specifically: either BootP server 232 or DHCP (dynamic host configuration protocol) server 234, and my inventive random access trivial file transfer protocol (RATFTP) server 236. The BootP and DHCP servers are conventional in

nature and, as such, will not be discussed in any detail. On the other hand, the RATFTP server, while based on and extends capabilities of a conventional trivial file transfer protocol (TFTP) server, accesses individual desired sector(s) (rather than just a complete file as does a conventional TFTP server), on hard disk 54 situated within server 50--thus facilitating client hard disk emulation. Such sectors are specified by a boot loader and downloaded into client PC during the network boot process.

(*Klimenko*, col. 7, lines 11-41) (emphasis added) The process shown by *Klimenko* does not provide for any request for a memory address region, and does not provide for the loading boot load data into the requested memory address region. Instead, what is described is a system in which a boot code is accessed from the NIC (network interface card) or from PC ROM BIOS (basic input output system). This boot code is then read into memory, and provides a network connection for remotely booting up the client PC.

The other cited portions of *Klimenko* are also irrelevant to the current claims. For example, for the element of storing boot image data in a memory the Examiner has cited to a sentence in the following paragraph:

> As shown, client PC 10 comprises input interfaces (I/F) 310, processor 320, NIC 360, memory 330 and output interfaces 340, all conventionally interconnected by bus 350. Memory 330, which generally includes different modalities, includes illustratively random access memory (RAM) 332 for temporary data and instruction store, diskette drive(s) (not specifically shown) for exchanging information, as per user command, with floppy diskettes, and non-volatile mass store 335 that is implemented through hard disk drive(s) 334, typically magnetic in nature. Should client PC 10 be implemented by "diskless" computer, then all disk drives, including both floppy diskette drive(s) and hard disk drive(s) 334, would be omitted. Regardless of whether client PC 10 contained a hard

disk drive or not, the client O/S, during its boot process, would be downloaded into RAM 332 and executed therefrom. As shown above in FIG. 2A, NIC 360 contains internal read-only memory 362, that stores network boot code 364. This code, as will be discussed shortly below, once downloaded into RAM 332 on power-up permits the NIC to establish a network connection to a remote server.

(*Klimenko*, col. 7, lines 11-41) (emphasis added) Thus, *Klimenko* is discussing downloading the client operating system, not network boot data. The "network boot code" is instead stored in internal read-only memory in the network adaptor.

***Haskins* Reference** – As was explained in the prior response, *Haskins* fails to teach or suggest the elements of claim 1 that are missing from *Klimenko*. *Haskins* deals with the different issue of a least call routing system, and specifically with regard to choosing which of a number of telephone routes will result in the lowest cost. The reference does appear to have any relevance to the booting of a computer system.

The Examiner has cited this reference specifically with regard to a memory address region. To the degree that *Haskins* relates to the request for data or a memory region (which is not conceded) this with regard to unrelated and incompatible technology and data. There does not appear to be any connection to a request for network boot data – the data requested appears to relate to rate provider information that may be used in routing telephone calls at the lowest cost. *Haskins* fails to teach or suggest the elements of requesting a memory address region and network boot data from a server, receiving the network boot load data and a designated memory region from the server, or loading the network boot load data into the designated memory region. *Haskins* relates to an unrelated technology, and does not appear to be relevant regarding any of these claim elements.

***Burokas* Reference** – The newly cited reference *Burokas* deals with a system and method for network booting of an operating system on one or more client devices. Again, the elements of claim 1 are not present in this reference. It is noted that *Burokas* actually deals with waking up systems that are in hibernation. In such a process, a hibernation image is saved in non-volatile memory, and resuming operation generally involves executing a resume function. In this particular technology, data that is necessary to allow a boot process to proceed, such as a hibernation file that includes the hibernation image and several OS files, may be synchronously streamed from a network server. (*See*, col. 2, lines 56-65) The streaming of the data allows the data to be sent once rather redundantly to each client. Further, even if it is assumed for the sake of argument that a request for a hibernation file containing a hibernation image is relevant, it is submitted that this process still does not include the request for a memory address region, the receipt of a designated memory region, or the storage of the boot image data into the designated memory region. It is thus submitted that none of the references cited include these elements.

Incompatibility of References – Even if it is assumed for the sake of argument that the three references separately teach or suggest all of the elements of claims (which is not the case), the references are not properly combinable.

As indicated above, *Klimenko* describes a system which is incompatible with a request for a memory address region and boot image data from a server. *Burokas* deals with a process for waking a system from hibernation, which involves obtaining and reading a stored hibernation image, and this is inconsistent with the process described in *Klimenko*.

Haskins relates to an irrelevant technology, but, even if this factor is ignored, the elements of *Haskins* could not be combined with either *Klimenko* or *Burokas*. The request for data would not make any sense with regard to the reading of ROM data in *Klimenko*. Further, *Burokas* deals with waking a system from hibernation, and the data requests described in *Haskins* relate to data transfers in operational systems, which would make no sense in the process shown in *Burokas*.

It is thus submitted that there cannot be any proper motivation for combining the references – the references cannot operate together because the concepts described in each reference are inconsistent.

Thus, *Klimenko*, *Haskins*, and *Burokas*, taken alone or in combination, do not teach or reasonably suggest all of the elements of claim 1, as amended, and claim 1 is allowable. It is submitted that the arguments presented herein also apply to independent claims 8, 12, 19, and 27, and thus such claims are allowable for similar reasons. The remaining rejected claims are dependent claims that, while having other independent reasons for allowance, are allowable as being dependent on the allowable base claims.

Conclusion

Applicant respectfully submits that the rejections have been overcome by the amendment and remark, and that the claims as amended are now in condition for allowance. Accordingly, Applicant respectfully requests the rejections be withdrawn and the claims as amended be allowed.

Invitation for a Telephone Interview

The Examiner is requested to call the undersigned at (503) 439-8778 if there remains any issue with allowance of the case.

Request for an Extension of Time

The Applicant respectfully petitions for an extension of time to respond to the outstanding Office Action should one be necessary. Please charge any fee to our Deposit Account No. 02-2666.

Charge our Deposit Account

Please charge any shortage to our Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: March 1, 2007

/Mark C. Van Ness/
Mark C. Van Ness
Reg. No. 39,865

12400 Wilshire Boulevard
7th Floor
Los Angeles, California 90025-1026
(303) 740-1980